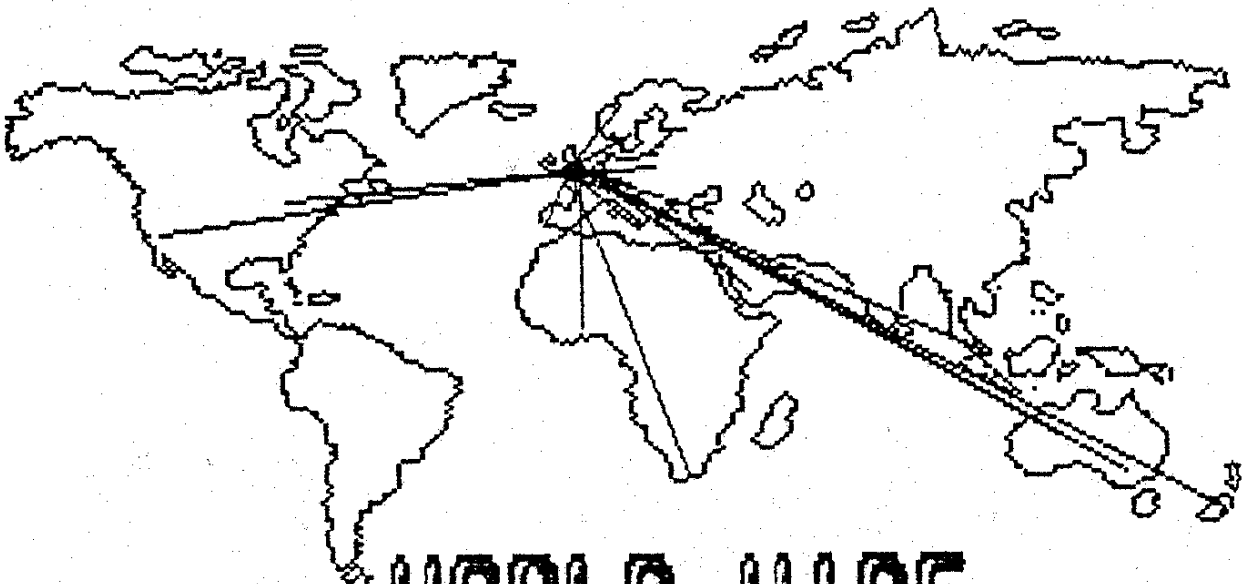


FORNAT

THE NEWSLETTER
of the
INDEPENDENT
DISCIPLE USER GROUP

ISSUE 1 AUGUST 1987



WORLD WIDE..
AND GROWING FAST.

INDUG.

CONTENTS

ISSUE 1 - AUGUST 1987

The Editors Page	3
News	4
Version 3 Review	5
Standards	8
Inside 2c	10
Open & Close	11
The Help Page	13
Writing/Next Month	15

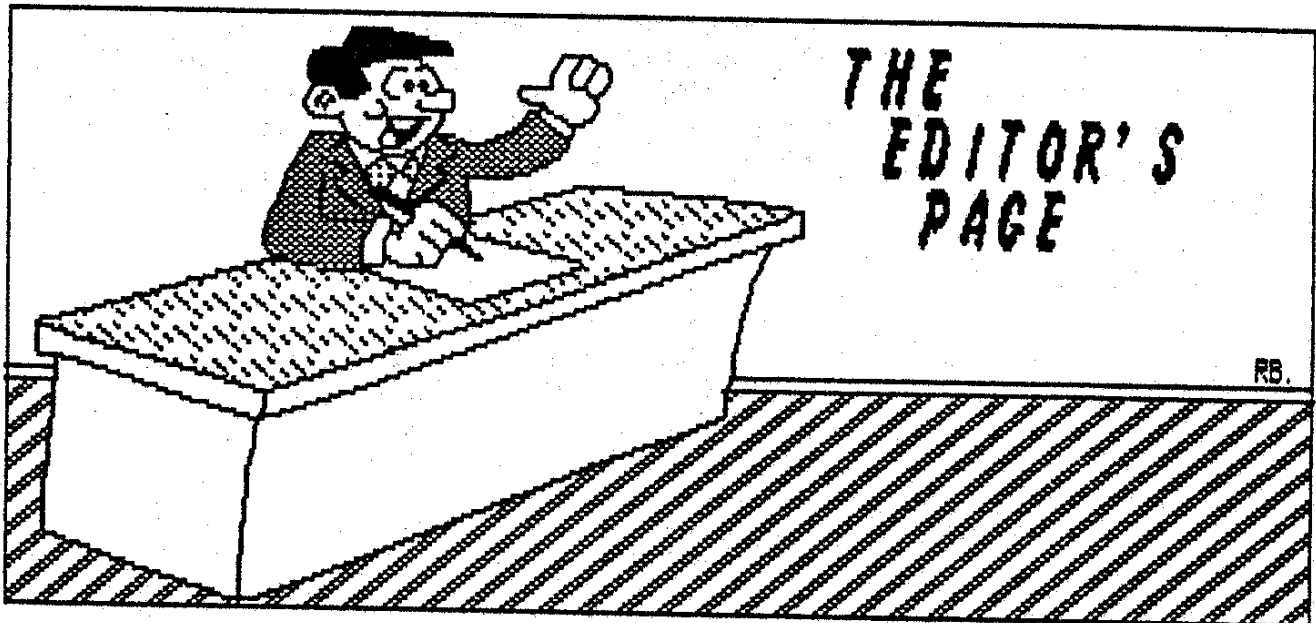
COPYRIGHT (C)1987 INDUG

No part of this newsletter may be reproduced in any form without the written consent of the editor. Members of the User Club may copy program items etc; but only for their own personal use and enjoyment.

FORMAT is published by:- The Independent DISCiPLE Users Group (INDUG) 34, Bourton Road, Gloucester, GL4 0LE, England.

DISCiPLE is a trade mark of Miles Gordon Technology.

The DISCiPLE interface is marketed by ROCKFORT PRODUCTS, 81 Church Road, London, NW4 4DP.



When I was planning the user group I really had no idea it would take off so well or so quickly, we already have members as far afield as New Zealand, Australia, USA, Africa and all over Europe. Dear old ZX Computing (now sadly missed by us all) jumped the gun, and published my name and address two months ahead of schedule, so while I was still fighting to get the introductory issue together I was already swamped with letters asking for details of the club.

I would like to apologize to anyone who wrote to me during that period and had to wait so long for a reply, my first priority was to get the introductory issue into print, but I did make sure everyones name was on the mailing list Rockfort were using. As most of you know Rockfort kindly sent out the introductory issue as part of their mail-shot announcing the availability of Version 3. Delays in finishing V3, followed by problems with their database, (well, they were using an IBM so what do you expect) meant that instead of you all receiving your copy in May most of you had to wait until well into July. Still my thanks go to all at Rockfort for their valiant efforts to get things out in the end.

Its difficult to know where to start with this issue. So much has happened since the introductory issue was written. Version 3 is now out and lots more software is being converted to use the power of the DISCiPLE. Existing Spectrum owners have been abandoned by AMSTRAD making the new +3 almost totally incompatible with everything before it, and giving only 175k of disc on-line at once. We will cover the +3 in a future issue just to show you what some poor users are going to get lumbered with.

In this issue you will find a review of the new version 3 ROM together with the first of a series of articles on the new commands this gives you. While I think most users will upgrade to the new version, there will still be interest in V2 for some time to come. I have no intention of neglecting 2c users and this issue carries a ROM address list just for them.

Bob Brenchley.
Editor.

DISCIPLE NEWS.

THE LAST WORD

Following the collapse of Saga, Nick Buckenham of Myrmidon Software has decided to revamp and relaunch 'The Last Word' his highly acclaimed wordprocessor. By combining the best ideas from several versions, and including most of the extras which have been developed over the last year, 'TLW' will be without doubt the most powerful and versatile wordprocessor on the Spectrum. The new version will include full DISCiPLE compatibility and should be launched at the next ZX Microfair in August.

WORLD WIDE DISCiPLE

Despite being the middle of summer, sales of the DISCiPLE are forging ahead. Over 4000 units have been produced so far and overseas sales have accounted for a reasonable proportion of these. Spectrum users as wide spread as Australia, Saudi Arabia, Poland, America, and Sweden have taken to the DISCiPLE in a big way.

Alen Miles, of Miles Gordon Technology the designers of the DISCiPLE, said "British sales are still climbing month by month, but it is the large sales to places like Holland and the Scandinavian countries that is most encouraging."

SPARKLERS FADE OUT

Creative Sparks Distribution Ltd, producers of the Sparklers Budget games, have finally called in the receiver. Involved in software and hardware distribution, as well as their budget and full price game operation, CSD operated out of large premises in Farnborough. After a somewhat checkered history the company were unable to survive in the cut-throat world software distribution has become.

Mikrogen of Bracknell, who were taken over by CSD less than twelve months ago, are also in liquidation. Mikrogen were old favourites in the Sinclair world having published software since the early days of the ZX81. They will be missed by many Spectrum owners, there's not many of the old guard left now.

THE DISCiPLE SHOW

The next ZX Microfair, at the New Horticultural Hall, Westminster, on Saturday 22nd August, will see the launch of the DISCiPLE show. Those of you who have attended previous Microfairs will remember the large 'Stage' area at the top of the stairs. On the 22nd most of this area will be taken over by the DISCiPLE with Rockfort Products sharing the area with many other companies producing software and hardware for the DISCiPLE. Bob Brenchley will be there for INDUG. It looks like an exciting show so don't miss it.

VERSION #3

WHATS IN THE NEW DISCiPLE ROM FOR YOU

It's here at last, VERSION 3, the upgrade most DISCiPLE users have been waiting for. All DISCiPLEs shipped after early June have V3 fitted as standard and existing DISCiPLE owners can obtain an easy to fit upgrade from ROCKFORT, just open the case, fit the new EPROM (no soldering needed) load the new GDOS system from tape and away you go.

The new system has many improvements over 2c but at the moment there is no new manual. The simple photocopy sheets provided with the upgrade do little to explain the powerful new commands but we start a new series this month that will explain them all in full, meanwhile this review should get you started.

Probably the most important new features are the OPEN# and CLOSE# commands allowing BASIC programmers to handle data files. Both commands work in almost the same way to their microdrive equivalents but like so many other things in the DISCiPLE they go even further. PRINT# lets you write to a file while the INPUT# and INKEY## commands enable you to read data from a file. These files are labelled OPENTYP Its interesting to note that any file, even a Basic program file, can be OPENED but you may not be able to make sense of whats in it. Another thing worth noting is that a MICRODRIVE type file can be opened but you will need to explore its format before you can use it. This is due to the way the DISCiPLE stores the data output to this type of file. All sector header data is stored along with Pre- and Post-amble bytes. Perhaps someone out there will write a utility to unravel these and turn them into OPEN type files.

Next, we come to the Snapshot facilities which have been much improved. One of the problems on older versions was that it was too easy to knock the Snapshot button by accident. If this happened, and there was no disc in the drive, or if the disc was full, or write protected then your program would crash. Now with V3 you must hold down the Caps Shift key and press the Snapshot button for it to work. When you do this the program is halted and the border displays fine stripes indication that the system is waiting for you to press a key. There are now five keys you can press:-

- Key 1 = Print screen copy, normal size.
- Key 2 = Print screen copy, A4 shaded.
- Key 3 = Snap screen to disc (SCREEN\$ file).
- Key 4 = Snapshot 48k program.
- Key 5 = Snapshot 128k program.

Both screen dumps now produce shaded prints but the normal size dump is a bit limited. Black, blue, red and magenta colours set a pixel while green, cyan, yellow and white do not. This can cause problems with certain mixes of colours on screen. In the A4 dump all colours are given their unique 3x3 dot pattern

selected from a table which could be redefined by the user. The dump is, of course, a little slow but the end result is well worth the wait. Expensive wallpaper, but sure to turn your friends green with envy. While on the subject COPY SCREEN\$ 2 will perform the A4 dump from within a program.

The 128k Snapshot needs you to confirm whether the screen has changed after the first part of the save. This is due to the 128/+2 having two screens but no way to detect by software which screen was in use when the Snapshot button was pressed.

On the subject of buttons the RESET button on the Spectrum+, 128 and +2, now does not clear the GDOS area when pressed once (but it does if pressed twice). There is, therefore, no need to reBOOT the system or do an OUT before continuing to use the DISCiPLE.

On the 128/+2 the COPY command would not work in 128 mode. This is caused by a bug in the 128k ROM and to get round this version 3 now uses SAVE d1"FILENAME" TO d2 to copy a file from disc 1 to disc 2. This is, I think, the only backward step I can find in version 3, it would have been nice to use COPY even if only in 48k mode.

The DISCiPLE has always been able to read and write data files from machine code but now you can do it from Basic as well. OPEN#, CLOSE#, PRINT#, INPUT# and INKEY\$# are all included in the new command set. See page 12 for a full explanation of file handling.

MOVE is another new command to the DISCiPLE. It works by moving a file, one sector at a time, to either another file or to a stream. If a file is OPENed as a write file MOVE can be used to copy the contents of a file into the one you are working on, using its stream number. You could then add data before CLOSEing the new enlarged file. Unlike microdrives there is no need to OPEN files when MOVEing file to file.

CODE files have also undergone a revolution with the advent of version 3. You can now create an AUTO-RUN file by saving the machine code like this:-

```
SAVE d1"MCODE" CODE 40000,3000,40512
```

This saves a file 3000 bytes long from location 40000 with an entry point of 40512. No more Basic loaders just LOAD d1"MCODE" CODE (or LOAD Pn) and the file auto-runs once loaded. Now why didn't Uncle Clive think of that in the first place?.

Also for machine code buffs the new EXECUTE file is a bonus. This takes up no user memory, it sits in the DISCiPLE's buffer area, but can only be one sector long. The following line saves a sector length file to disc from location 60000.

```
SAVE d1"MYCODE"X,60000
```

To load your routine simply LOAD d1"MYCODE"X and in comes the code. The DISCiPLE then CALLs location 7126. Your routine should do a RET to get back to GDOS and should stack any registers you intend to use. So if your routine is less than 510 bytes long (254 in single density systems) and can be assembled to run at location 7126 (1BD6 hex) this is just the ticket for you.

Version 3 will allow the directory to be sent to the printer with CAT#3;DrvNo. If the short CAT is called the full width of

the printer is used. Due to a bug in the ROM you can't CAT to any stream connected to a disc file, I will try to find a patch for this in the future.

We will now look at two new commands which deserve the following message:-

WARNING: THESE COMMANDS CAN DAMAGE YOUR SANITY

The commands LOAD@ and SAVE@ read and write sectors to disc. If used by the unwary it is only too easy to overwrite the wrong sector and corrupt your disc. I would strongly recommend that you back-up your disc and work on the copy, then if you make a mistake all is not lost.

```
LOAD @ D,T,S,Location
SAVE @ D,T,S,Location
```

were D = drive number, T = track number and S = sector number and Location is an address in memory to which the command refers. LOAD@ 1,0,1,40000 will load track 0, sector 1 into memory at location 40000. This is by the way the first sector of the directory. If double density 512 bytes are loaded or saved, in single 256 bytes

On the DISCiPLE track numbers start at zero on side 1 of the disc and at 128 on side 2. Sector numbers are from 1 to 10. LOAD@ and SAVE@ are probably the most powerful commands in the DISCiPLE system, use them with care.

The NETWORK has been expanded with several new features and quite complicated mixes of DISCiPLEs, with or without disc drives, with or without printers, and even Spectrum with interface 1 can be linked together.

Finally a few little extras have been included. CLEAR# will clear the streams area and reset streams 4-15 to zero. CLS# will set BORDER & PAPER to 7, INK, BRIGHT, FLASH & OVER to 0. It then clears the screen. POKE@ now has new uses. 'RUN boot' will boot your system without loading the AUTOLOAD file. The SYStem file can now be anywhere on the disc and the first filename starting with "SYS" is loaded. The last seven letters can be used as a name for the disc. FORMAT d2 TO d1 formats disc 2 and copies, sector by sector, the disc in drive 1. DO NOT USE this on a single drive system as there are no prompts to change the discs over.

There is so much in the new system that any review can only hope to scratch the surface. The question I am most asked is "Is version 3 worth the price of the upgrade (about £9)". My answer is a very definite yes. For the games player there is the extra Snapshot and screen dump facilities. For the serious user OPEN & CLOSE are in themselves worth the upgrade. There are still a few things I would like to have seen included in the DOS but I can assure you there is no room left whatsoever.

Bruce Gordon, the designer of the DISCiPLE and author of GDOS, must be congratulated on producing such a powerful and versatile system.

I look forward to bringing you more details over the next few months, but first turn to page 11 and find out all thats worth knowing about OPEN and CLOSE.

A MATTER OF STANDARDS

When storing programs on tape its usual to use short C10/C15 tapes and save one program per side. On microdrive there was still only room for a handfull of programs on each cartridge. Now, with the advent of the DISCiPLE, you can have up to 80 files per disc in double density mode. With so many files you really need to adopt some form of standard in the way you name each file.

Now to lay some ground rules, the DISCiPLE allows filenames of up to ten characters, no two files on the disc can have the same name and filenames should not contain the characters "*" or "?" unless you are looking for problems in loading them back. The DISCiPLE gives a very full and detailed CATalogue listing but this is only useful if you have less than 20 files on the disc. With more files you can only see part of the list at once. The short CAT only prints the filenames but at least gets most of them on the screen at once.

If we can combine a standard for naming our files with a method that will enable us to make better use of the short CAT function we will be on to a winner.

Any method we adopt must be able to deal with a wide range of file types and situations. It must deal with Basic, Code, Screen\$, UDG, Data, Array and Text type files, just to name the most popular.

Lets start on the road to good standards with one of the easiest changes, one which will save you much time and effort in the future. For the purpose of this article we will consider a program called "PACMAN", this consists of three files, a Basic program, a SCREEN\$ file and a CODE file holding the machine code.

Our first priority is to ensure all three parts of the program are identifiable as related to each other. The second priority is to distinguish between each part.

If we return to our example we can start off by saving the Basic program as "PACMAN". We now need to save the other parts in a way that will ensure we know what they are and which program they relate too. Lets assume we can always think of a name for our program which is only 8 characters long (or less). Not too difficult, the first ICL main-frame computer I worked on only allowed 4 character names for programs, so 8 characters is a positive luxury.

We can then use the other two characters to indicate the file type but to make things look good lets just use one character and separate it from the main name with the underline character (symbol shift/zero).

To aid memory we will use "_S" for the SCREEN\$ file and "_C" for the CODE file.

So our three files will now be called "PACMAN", "PACMAN_S" and

"PACMAN_C" making it very clear what they are. Most disc systems use some form of file extension in this way and it is an easy system to remember.

If there were two CODE files, or even more, to this program then we could name them "PACMAN1_C" and "PACMAN2_C" etc, so indicating which order they are to be loaded in. You will have noted that Basic programs have no extension under this system so they are still set aside from the rest.

Extending these ideas would give us the following table

"_C"	- Code Files
"_S"	- SCREEN\$ Files
"_N"	- Numeric Arrays
"_\$"	- String Arrays
"_D"	- Data Files (Open-type)
"_G"	- Graphics (UDG) Files
"_T"	- Text Files (Tasword ect)
"_F"	- Font Files

OK so far, your directory listing looks good and you can find the program you want (and its associate file) quickly but there are a few more things these standards are useful for.

Take the COPY command, to copy our PACMAN program from one disc to another all we need to do is COPY d1"PAC*" TO d2 and all three files are copied by one command. The same applies to ERASE and CAT. CAT 1;"P*" will list all files starting with the letter 'P' to the screen.

Now lets break our own standards, WHAT so soon after laying them down I hear you say. Well yes, this standard is excellent for bringing together the parts of a program but there is one circumstance where a slight alteration is very useful. I refer to programs such as Tasword where a disc may contain dozens of files all created by the same program. Obviously with a wordprocessor you need to give a meaningful name to your text files but even with an extension of "_T" things are not perfect. Unless all your names are exactly eight characters long the command CAT 1;"????????_T" will not work.

If however we start with "T_" and then follow it with the name things get a bit better. We can now CAT 1;"T_*" to get all our text files listed. COPY d1"T_*" TO d2 would copy all the text files on disc 1 to disc 2. ERASE d1"T_Fred*" would erase all the letters we wrote to Fred.

Another standard to bear in mind, especially if you have a two drive system, relates to loading files from within a program. If your Basic loader program contains the line LOAD d1"PACMAN_C"CODE then the disc must always be in drive 1. It's much better to say LOAD d*"PACMAN_C"CODE so that it works regardless of the drive the disc is in. A small point but it does allow full use of both drives.

CPM, MSDOS and all the rest force you to use file extensions, GDOS does not but that's no excuse not to get into the habit.

With prefix or extension codes coupled with meaningful names, most, if not all, of the problems encountered with large disc systems can be overcome. Each user can design his own standard, tailored to his own individual needs, but the need to adopt some form of standard is clear.

INSIDE GDOS Y2c.

For those of you still using version 2c of the DISCiPLE operating system GDOS here is some information on addresses in the shadow RAM/ROM area.

ADDR	HEX	BYTES	CONTENTS
8	0008		Address at which GDOS pages in when an error is found in main ROM.
16	0010		RST 16 routine. Calls routine in main ROM. Use RST 16, DEFW address to call.
79	004F		Return to main ROM routine. PUSH the address you want to jump to onto the stack and jump to location 79. this pages in the Spectrum ROM, does an EI followed by an RET.
94	005E	2	Length of Snapshot. Normally 49152.
96	0060	2	Start address of Snapshot. Normally 16384.
102	0066		NMI routine. Start of Snapshot.
543	021F		HOOK Code table. Two byte addresses starting with Hook Code 51 (33 hex)
583	0247	10	Name of autoload file. Set to "AUTO* " but could be set to any name you want
654	028E		This routine is paged in every time the keyboard is scanned. It looks at the network. If a call was inserted here your routine would be called 50 times a second.
768	0300		Start of GDOS system variables POKE @0 = 768
782	030E	2	On Error.(POKE@ 14 & POKE@ 15) if these two locations are poked with the two byte address of your routine then it will be called if a Basic line fails both Spectrum and DISCiPLE syntax. Do a RET to get back into the DOS, remember the shadow RAM/ROM is paged in.
826	033A	24	Character definition for £,# and (c)
4063	0FDF		COPS1. This is the routine called by Snapshot to do the screen copy. it ends at 4199 (1067 hex). If you need a different screen copy or a routine to operate on snapshot this is the place to put it.
4693	1255		System message area.
5015	139F		Error Messages. No fixed length, the first message starts with CHR\$ 0, the second with CHR\$ 1 ect. Could be used to provide your own error messages. NOTE in GDOS 2c error messages are displaced by one. too correct this bug POKE@ 4177,0 and resave your system.

That all for now. If anyone has worked out any others please let us know.

AN OPEN & CLOSE CASE

The statement `OPEN#4;d1"test"` will open a file called "test" on drive 1. If the file already exists it will be opened as an input file, if the file does not exist then an output file is created. On microdrives if you already had a file of the same name on the cartridge you had to ERASE it before OPENING it if you wanted to write a file, a messy way of doing things. In V3 the DISCiPLE adds two extensions to the OPEN# syntax, IN and OUT (using the BASIC keywords). `OPEN#4;"test" OUT` will force the file to be opened as a write file and if it already exists will prompt you with the OVERWRITE Y/N question which by the way now prints the filename as part of the prompt. `OPEN#4;d1"test" IN` opens a read file, if the file is not found an error message is given. You can have up to 12 files OPEN at once (Streams #4 to #15) and, like Interface 1, each file has a buffer in the Channels Area so the more files you have the more memory they take up.

Due to the way the DISCiPLE keeps track of free sectors when writing files to disc you can only use one drive at a time for OUT files although you can have more than one OUT file on the same drive. There is no such restriction on IN files as the DISCiPLE does not need to keep a record of free space for these.

Note that unlike microdrives you can't redirect stream #3 (the printer stream) to an output file.

Having OPENED a file you need to be able to do something with it of course. `PRINT#n` will write data to Stream n if the file is an OUT file, `INPUT#n` or `INKEY$#n` will read data from an IN file.

Once you have finished with a file you will need to CLOSE it. `CLOSE#*4` will close the file OPENED on stream #4, if it was a write file the current buffer is written to disc, the disc directory is updated with the files details and the Channel Area allocated is recovered. Once a stream is CLOSED its number can be used again if required. `CLOSE#*` without a stream number will close all current files, any data in the buffers for output files will be written to the disc. To prevent corrupt (unclosed) output files on the DISCiPLE, the RUN command does not clear the Channels. You must CLOSE files to free the Stream number.

When an OUTPUT file is used the directory entry is not made until the `CLOSE#` command is issued. It is therefore good practice to do a `CLOSE#*` if your program crashes to preserve any data you have already written.

OK so lets see this in action, type in the following short program and try it out.

```
10 OPEN#6;d1"testfile" OUT
20 FOR I=1 TO 10
30 INPUT "Type in a number please ";NUM
40 PRINT#6;NUM
50 NEXT I
60 CLOSE#*6
```

```

70 OPEN#6;d1"testfile" IN
80 FOR I=1 TO 10
90 INPUT#6;data
100 PRINT "For input ";I;" you entered ";data
110 NEXT I
120 CLOSE#*6

```

Line 10 opens an OUT file. Lines 20 to 50 ask you for a series of numbers and then writes them to the file. Line 60 closes the file and frees the stream. The rest of the program opens the same file as an IN file, reads the number you entered from disc and prints it to the screen. A bit boring I know but it does demonstrate the principle.

Files created by an OPEN# command appear in the full directory list as 'OPENTYP' and with up to 780k of disc space you can now store an awful lot of data.

The microdrive syntax OPEN#n;"M";drive;"FILENAME" will work, although you lose the benefit of the IN or OUT extensions, but CLOSE# must have the * inserted to fail the Spectrum ROM syntax which has a fatal bug in it.

Right that's the simple stuff over with now lets deal with some more advanced matters. It would be useful not to read beyond the last item of data on an IN file. This would avoid the 'End of File' message which would stop your program.

The secret to avoiding this problem hides away in the channel area created when the file is opened. Three bytes, originally stored in the directory when the file was written, hold the High, Middle and Low bytes of a count of the number of characters in the file. The following subroutine returns the number of characters left on the file, if this is zero then any further attempt to read from that file will give the EOF error.

```

2000 REM enter with STN=stream number.
2001 REM exit with CL = characters left
2010 LET OFFSET=PEEK (23574+STN*2)+256*PEEK (23575+STN*2)
2020 LET CHANADR=PEEK (23631)+256*PEEK (23632)+OFFSET-1
2030 IF CHR$(PEEK (CHANADR+4))<>"D" THEN PRINT"Not a disc
file":STOP
2040 LET CL=PEEK (CHANADR+31)+256*PEEK (CHANADR+32)+65536*
PEEK (CHANADR+18)
2050 RETURN

```

Now add this line to the test program given earlier and RUN it.

```

105 LET STN=6:GOSUB 2000:PRINT CL;" Chrs left in file."

```

Line 2030 tests that stream #6 is attached to a disc file Notice that the count of characters include the carriage return at the end of each field.

If you need to keep a check on the number of characters written to an OUT file the count is stored in CHANADR + 231 (low), 232 (middle) and 219 (high).

The routine could be adapted to cope with both IN and OUT files. Location 9 + 256 * Location 10 will equal 551 if the file is an input or 787 if it is an output file.

I hope this article has helped you to understand how OPEN type files are used. I think we will be returning to them many times in the future.

THE HELP PAGE

Problems with your DISCIPLE? Dont worry, write to the HELP page Remember to quote your membership number and leave the problem to us.

NO DISC

How do you stop the disc drive motor without resetting the computer? i.e. following the "NO DISC IN DRIVE" error message the motor just keeps spinning.

D.SMART London.

This is a fault common to most disc systems, even the BBC micro suffers from exactly the same problem. I'm glad to say that the answer is very simple, if a disc is inserted into the drive in the normal way it will spin for a few seconds and then the drive will stop. As the heads are not loaded at this time you will not damage your disc or drive.

PRINTER OFF

Having used the DISCIPLE's printer port I need to be able to switch back to the Spectrum 128 RS232 port. Is there any way, other than RAND USR 0, that I can do this.

PAUL SMITH. Devon.

If when you set up your SYSTEM file you said you would be using the DISCIPLE's printer port then this is selected when GDOS boots in from disc. GDOS sets the pointers in the channel area to use its routines. The Sinclair ROM never resets this area except when you do a NEW or reset the machine. To use the 128's RS232 or a ZX printer on the 48k you must overwrite the channel pointers GDOS set up. Run this short program, with the correct DATA line for your machine to recreate the original Spectrum channels.

```
10 POKE @11,1:REM switch of DISCIPLE printer.
20 LET A= PEEK 23631+256* PEEK 23632
30 FOR I=A+15 TO A+20
40 READ X:POKE I,X
50 NEXT I
60 DATA 244,9,196,21,80:REM 48k
60 DATA 52,91,47,91,80:REM 128k
```

To return to the DISCIPLE's printer just POKE @11,0.

COPY PROBLEM

The single disc COPY command does not seem to work properly on

my V3 DISCiPLE. I send the command to copy a file and the file is loaded from the disc then I get the CHANGE DISC message. With the new disc in the file is saved but I then get the CHANGE DISC message again. Whats going on? and why does the system NEW itself at the end.

J.F.Hamilton. Petersfield.

From the way you describe the COPY command everything seems in order. The routines that handle COPY syntax are set up to allow for multiple files to be copied. The extra change disc is so that it can continue down the directory list on your source disc to see if there are any other files to be copied. The NEW is because the spectrums RAM may have been corrupted by the COPY so it must reset the memory. If you need to copy a file without the system NEW, use the MOVE command but this is not really suitable for single drive systems.

ARTIST II

Inserting 'PRINT@6,1' at line 90 of the basic header and changing the code for the print routine to that of the KEMPSTON 'E' enables ARTIST II to work a treat with the DISCiPLE system. However I can't get the joystick option to work properly. It will work on exit from the initial windows but not before. Any ideas?

ALAN COUTTS. Oakham

I am not really familiar with the ARTIST II program but I think I know where the problem lies. The DISCiPLE's joystick port mimics both KEMPSTON and SINCLAIR ports. If the program is scanning the KEMPSTON port address it may still be detecting the key-presses caused by the SINCLAIR port. There is no certain way round this but if the program gives an option for SINCLAIR joystick this could solve the problem.

TWO DRIVES

I only have one disc drive at the moment but would like to upgrade to two drives. How can I connect two drives to one connector or will I need to sell my drive and buy a twin drive system.

D.CULL. Chichester.

No problem, all you need is a converter which takes the two plugs on the end of the ribbon cables and gives you one plug to go into the DISCiPLE. Watford Electronics (tel 0923-37774) have advertised one in the past, or look in one of the BBC magazines, there's bound to be at least one advert for an adaptor.

Thats all I can fit in this month. If you have an urgent problem remember the INDUG 'Hotline'. Please be patient if you write in, letters with a SAE will get a reply but I am a little behind at the moment.



WRITING FOR FORMAT

Contributions from FORMAT readers are very welcome. We like to publish articles on any subject relating to the DISCiPLE, the Spectrum or indeed any aspect of computing that you feel may be of interest to other FORMAT readers.

Some points to bear in mind

- * Ideally submit your article as a Tasword (2 or 3) or similar text file (with a printed copy). We can accept disc (5.25 or 3.5), tape or microdrive.
- * Mark everything with your name, address and telephone number.
- * Keep a copy, DO NOT send your only version, the post office is not that good (and nor are we).
- * Include a stamped addressed envelope if you want your material returned.
- * Remember to say which version of GDOS your program/article was written for, we will check other versions.
- * Feel free to contact us with your ideas before committing yourself to writing that long article.
- * DO NOT COPY ITEMS FROM OTHER MAGAZINES.

Remember we pay for all items published in FORMAT so get writing.

NEXT MONTH IN FORMAT. .

Next month will see the continuation of our series on the new commands in version 3 of GDOS. We will welcome John Wase (ex ZX Computing DISCiPLE Data Page) with the first in an irregular column for FORMAT.

We also take a look at the POKE@ command and explain some aspects of this command the manual omitted to tell you about.

SEE YOU NEXT MONTH.

SUPER DISCS

FORGET THE REST

BUY THE BEST

EXCLUSIVE TO INDUG MEMBERS

5¼ inch DS/QD 96tpi 100% ERROR FREE
10 Discs
in plastic case *only* **£14-95**

3½ inch DS/QD 100% ERROR FREE
10 Discs
in plastic case *only* **£19-95**

5¼ inch DS/DD 96tpi
~~BULK PACK 25 DISCS~~ **£19-95.**
Duplication Standard

SEND CHEQUE/P.O. IN STERLIN TO

NATIONAL DISC SUPPLIES
16 Westland Rd, Hardwicke,
Gloucestershire, GL2 6QH.

ALL PRICES INCLUDE UK POST AND PACKING.
OVERSEAS ADD £3 EUROPE ADD £1.50.